SPECIFICATION

Please amend the last paragraph on page 8 as follows:

The present invention facilitates flexible consolidation and correlation of intrusion detection information from a variety of different Intrusion Detection System (IDS) sensors and systems. The present invention is capable of leveraging network and application management platform (e.g., OVO) features to provide improved and effective newly deployed IDS solutions in a cost effective manner. The present invention also reduces resources required to coordinate and implement an effective enterprise network and host intrusion detection system.

Please amend the first paragraph on page 9 as follows:

Figure 1A is a block diagram of intrusion detection integration system 101 in accordance with one embodiment of the present invention. Intrusion detection integration system 101 includes intrusion detection integration console 102 and user interface 109. User interface 109 permits a user to interface with intrusion detection integration console 102. User interface 109 can include a display (e.g., a [[VGA]] Video Graphics Array monitor, flat panel monitor, etc.) and an input/output device (e.g., a keyboard, mouse, etc.). Intrusion detection integration console 102 consolidates various different types of intrusion detection information from a variety of different types of intrusion detection sensors and systems.

Please amend the third paragraph on page 12 as follows:

In general, UDC 100 includes a programmable infrastructure that enables the virtual connection of selected computing resources as well as the isolation of selected computing resources, thereby enabling security and segregation of computing resources at varying infrastructure levels. The resources included in UDC 100 can be dynamically programmed to logically reconfigure and "separate" the resources into a number of various virtual local area networks (VLANs). In one exemplary implementation, Network Operations Center 170 (NOC) [[170]] includes server 171 coupled to a user interface 191 and a utility database 192.

Please amend the paragraph that starts on line 25 of page 12 and ends on line 11 of page 13 as follows:

Utility controller database 192 comprises configuration information pertaining to the various resources in UDC 100, including descriptions of the configuration, characteristics, and/or features of a component. For example configuration

information can include but not necessarily be limited to indications of the types of devices in UDC 100, representations of each VLAN, a network or MAC (media access control) address for the resources of UDC 100, port numbers of the configurable components, VLAN identifiers associated with each of the port numbers, socket identifier for each cable connected to each of the resources of UDC 100, manufacturer identifiers, model indicators, and/or serial numbers. As resources in UDC 100 are changed (e.g., reallocated), the information in utility controller database [[150]] 192 is also changed accordingly (e.g., to reflect the reallocation). Changes to the utility controller database [[150]] 192 can also be used to drive changes to the allocation of resources in UDC 100.

Please amend the paragraph that starts on line 21 of page 13 and ends on line 3 of page 14 as follows:

Server 171 includes a network application management platform [[172]] 173 (e.g., an open view operation network application management platform) for managing resources in UDC 100 in accordance with information included in utility database 192. For example, utility controller 172 enables the creation, deployment, allocation, and management of VLANs. In one exemplary implementation, utility controller 172 can monitor deployed VLANs, and automatically reallocate resources when there is a reason to do so. In addition, the utility controller 172 monitors shared infrastructure resources, alerting NOC 170 of failures or other significant events. Utility controller 172 utilizes network application management platform 173 to manage resources in UCD 100.

Please amend the second paragraph on page 14 as follows:

Network application management platform [[172]] 173 also includes an intrusion detection integration console 174. Intrusion detection integration console 174 consolidates intrusion detection alarms from various sensors and IDS systems. For example, intrusion detection system consolidation console 174 consolidates information from NIDS sensors 121 through 126 and HIDS sensors 137 through 139 and 147 through 149. IDS consolidation console 174 facilitates centralized management of multi-vendor and multi-type sensors with minimal user effort. Intrusion detection integration console 174 provides a centralized alert logging wherein the alerts are consolidated and standardized in the severity assignment and at the same time correlated based on various alert attributes. The correlation of events by intrusion detection integration console 174 helps to provide a high level of

confidence in the intrusion alerts by reducing the probability of false positives and false negatives (e.g., beyond what is already done by an individual vendor IDS sensor engine). The correlation also facilitates automated configuration of reactions for the alerts based on various factors, including conforming the reactions to a standard enterprise response strategy.

Please amend the first paragraph of page 16 as follows:

In step 210, information from a plurality of different types of intrusion detection sensors is gathered. The present invention is compatible with a variety of interfaces provided by different vendors for network or IDS components to log new alerts to a third party product. For example, mechanisms can include a Simple Network Management Protocol (SNMP), syslog or an Application Programming Interface (API) through which the alerts are pushed out as they are raised. In one exemplary implementation, a HIDS provides the API, a NIDS provides SNMP traps, while the network devices log to a system log (e.g., syslog). The present invention is also able to react to a device or system that includes more than one way to provide the alerts to an external interface by analyzing each of the ways and selecting the method that is the most secure and has fewer dependencies in its communication path. For example, the API can be a preferred method because with the help of management application network platform (e.g., OVO) agent features it is possible to ensure that an alert does not leave the system on which the alert is detected. Alternatively, even though syslog approach and SNMP can be equally secure, SNMP can be more desirable than the syslog approach because it is less difficult to scale and less accessible for use by rogue applications that can embed false alerts into the syslog. In addition, a determination can be made whether the same channels are used by an IDS or network device to log any specific errors, so that those errors can also be channeled to a central repository.

Please amend the first paragraph on page 18 as follows:

Referring still to step 220 of Figure 2, standardizing the format for incorporating the IDS alerts into templates includes information related to correlation attributes and incidence response attributes of the alarm in one embodiment. For example, alarm correlation information can include a date and time stamp (which can be assigned by default), severity, component or sensor name (e.g., [[Dns]] Domain Name Server name, [[IP]] Internet Protocol address, firewall interface name, etc.) and type (e.g., NIDS, HIDS, firewall, etc.), attacker details, signature type or

attack details, and attacked victim host details. The alarm incidence response attribute information can include cause information (e.g., what is the root cause of an alert), recommended action(s) for an operator to take (or alternatively provide a document that explains the response strategy policy framework and provides actions to be taken for attacks of differing severities), automatic reactions configured as a response, and references to more detailed information about the alerts (e.g., the IDS [[GUI]] generalized user interface console specifics, or a pointer to a document / site that explains the attack detected). In one exemplary implementation, the messages use consistent message terminology.

Please amend the paragraph that begins on page 16 of page 21 and ends on line 6 of page 22 as follows:

In one embodiment of the present invention, intrusion detection method 200 also facilitates management of detection sensors. For example, the application feature of an OVO platform can be used for centralized IDS sensor management in which an authorized operator is able to access different sensor resources. In one exemplary implementation, one operator is able to read an IDS configuration file, while another operator is able to actually re-configure the file remotely, including starting and stopping an IDS sensor process after the reconfiguration. In one exemplary implementation, a variety of techniques (e.g., [[NNM]] network node manager, SNMP trap handling, monitor templates, etc) can be utilized to detect a sensor is operating (e.g., "alive") and monitor specific performance metrics via a scheduled script (e.g., checking the state of the IDS sensor process, its resource usage, its memory usage etc). An OVO platform can also utilize a template to schedule work on IDS sensors (e.g., backing up the evidence logs and creating fresh logs). Thus, the features of an OVO platform can facilitate effective management of a sensor via a centralized console. In addition, centralized policies regarding the management issues can be uniformly enforced via these mechanisms (e.g., backing up evidence logs across sensors every hour across the infrastructure).

Please amend the second paragraph on page 22 as follows:

Figure 3 is a block diagram of computer system 300, one embodiment of a computer system on which a present invention intrusion detection central system can be implemented. For example, computer system 300 can be utilized to implement intrusion detection integration console 174 or integrated intrusion

detection method 200. Computer system [[350]] 300 includes communication bus 357, processor 351, memory 352, input component 353, bulk storage component 354 (e.g., a disk drive), network communication port [[357]] 359 and display module 355. Communication bus 357 is coupled to central processor 351, memory 352, input component 353, bulk storage component 354, network communication port 357 and display module 355.

Please amend the paragraph that starts on line 38 of page 22 and ends on lin3 5 of page 23 as follows:

The components of computer system 300 cooperatively function to provide a variety of functions, including performing emulation application revision in accordance with the present invention. Communication bus [[307]] 357 communicates information. Processor 351 processes information and instructions, including instructions for coordinating security information from a plurality of different security intrusion attempt identification components. For example, the instructions include directions for integrating (e.g., consolidating and correlating) IDS information. Memory 352 stores information and instructions, including instructions for coordinating security information from a plurality of different security intrusion attempt identification components. including, integrated IDS information. Bulk storage component 354 also provides storage of information. Input component 353 facilitates communication of information to computer system [[350]] 300. Display module 355 displays information to a user. Network communication port 357 provides a communication port for communicatively coupling with a network.